A Simple MPLS-based Flow Aggregation Scheme for Providing Scalable Quality of Service *

Srinivas Vutukury
Computer Science Department

J.J. Garcia-Luna-Aceves
Computer Engineering Department

Baskin School of Engineering
University of California, Santa Cruz, CA 95064
{vutukury,jj}@cse.ucsc.edu

Abstract

Providing Quality of Service (QoS) capabilities in IP networks is the focus of much research in the Internet community today. The Intserv architecture proposed by IETF for supporting QoS in IP networks is known not to scale well to large networks due to the per-flow mechanisms it uses. While connection-less approach has been the central design paradigm of IP networks, many researches today believe connection-oriented mechanisms similar to virtual-circuits in ATM must be incorporated in IP networks in order to provide scalable QoS. Towards this end, the MPLS (Multi-Protocol Label Switching) technology has been put forward by the IETF. A crucial scaling issue with MPLS is that large number of LSPs (Label Switched Paths) may be required in the routers resulting in increase of state size in the routers. In this paper, we introduce the notion of *Label-Switched Multipaths* (LSMPs) and propose a simple technique for aggregating LSP into LSMPs such that the number of labels required in the routers is significantly reduced. Based on LSMPs we describe an architecture for providing deterministic guarantees that is far more scalable than architectures based on simple LSPs or those that use only multipoint-to-point LSP aggregation.

1. INTRODUCTION

Today there is great demand to extend the best-effort service of IP networks to provide service classes that support QoS-sensitive real-time multimedia applications. To address this demand, the IETF proposed several architectural frameworks such as Intserv,[4] Diffserv[6,10] and Traffic Engineering.[2] A key underlying requirement in all these frameworks is a mechanism to "pin routes" so that packets of a flow travel along the same path from the source to the destination. In recent years, the *Multi Protocol Label Switching* (MPLS)[14,5,15] has received tremendous attention as a means of pinning routes in IP networks. In MPLS, a set of paths is determined based on the factors such as QoS, policy routing and traffic engineering. Then *label-switched paths* (LSPs) are set-up for the paths using labels in the routers. Though MPLS-based packet forwarding is fast, simple and flexible, an MPLS-based architecture cannot scale if large number of paths need to be setup in the network. To reduce the label state-size, the LSPs must be aggregated. However, to date, apart from multipoint-to-point LSP aggregation, there have been no proposals for more sophisticated LSP aggregation. In this paper, we introduce a LSP aggregation method based on the notion of *label-switched multipaths* (LSMPs) and show how LSMPs can significantly reduce the label state size contributing to the scalability of the above MPLS-based frameworks.

Although there have been several proposals on using MPLS with Diffserv architecture and Traffic Engineering, surprisingly there has been no concrete proposals to use MPLS in the context of providing scalable deterministic guarantees services. The main reason being that setting up explicit route on per-flow basis is complex, and if flows are to be aggregated to reduce complexity, providing deterministic guarantees becomes difficult. We show that combining LSMPs with flow classes have the potential to yield a scalable architecture that provides guaranteed services. Recently, we proposed the SMART architecture for supporting deterministic QoS in IP networks using a purely connection-less approach. We adapt some of the techniques introduced in the SMART architecture to the proposed MPLS-based architecture. Combining LSMPs and flow classes simplify several mechanisms such as link-scheduling, resource reservation management, QoS routing and signaling. For example, to achieve scalability in link scheduling, we use special techniques to aggregate flows into flow classes such that the delay bound offered toa a flow at a link is the function of class and link characteristic and is independent of the number of flows that flow through the link. For maintaining resource reservations in the routers, a soft-state approach like the one used in RSVP[19] offers an elegant fault-tolerant solution. However, if the reservation and routing state is high, then the periodic refresh messages

*This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under grants F30602-97-1-0291 and N66001-00-1-8942.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **A Simple MPLS-based Flow Aggregation Scheme for Providing Scalable Quality of Service**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**7**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

used by the soft-state mechanism may prove costly. Also most traditional architectures require periodic link advertisements and multi-constrained path selection, both of which are not scalable. The application of LSMPs and flow classes alleviates these problem. In summary, the contributions of this paper are two fold. First, we describe a technique for aggregating LSPs into LSMPs. Second, we show how LSMPs can be used in the specific context of providing deterministic delay and bandwidth guarantees.

The paper is organized as follows. Section 2 describes our aggregation techniques for LSPs and introduces the concept of LSMPs. Section 3 gives some experimental data on the degree of aggregation achieved using LSMPs. Section 4 describes a scalable QoS routing architecture based on LSMPs. Section 5 concludes this paper.

2. LABEL-SWITCHED MULTIPATHS

Route-pinning in IP network is implemented using MPLS.[14,5,15] Under the label-switching scheme, the packets are assigned labels with specific meaning to the receiving router. In the most basic architectural model based on MPLS, a set of loop-free paths is determined for each source-destination pair Constraint-Based Routing and Traffic Engineering. Then *label-switched paths* (LSPs) are set-up for the paths. The core forwarding mechanism in label-switching is that a router uses the label embedded in the packet to determine the action that must be taken, such as the kind of QoS that must be provided, and before forwarding the packet to the next-hop router, the router replaces the current label with a new label. The same process is repeated at every downstream router until the packet eventually reaches the destination. This simple label-switching framework provides a simple yet powerful mechanism for fast packet forwarding. However, this approach can be expensive as the set of paths increases as is the case, for example, when finer granularity in QoS is required or greater utilization of network through traffic engineering is desired. Though the number of labels can be reduced by LSP aggregation, not much attention has been given to finding new methods for aggregating LSPs. The *multipoint-to-point* aggregation proposed to date is inadequate because the number of labels required in the routers can be still high. To address this problem we will introduce the notion of *label-switched multipath* (LSMP) that significantly reduces the amount of label state-size in the routers.

We will first briefly describe multipoint-to-point LSP aggregation and extend it to LSMPs. Multipoint-to-point aggregation is based on the rule that if two packets received by a router follow the *same path* starting from the router to the destination, then they must have the same label. The LSMP aggregation is a generalization of multipoint-to-point LSP aggregation and is based on the rule that if two packets received by a router follow *any of the paths in the same set of multiple paths* starting from the router to the destination, then they must have the same label. Packets with the same label may belong to different flows from different sources due to aggregation, but the routers process them identically based only on the label. We how describe how labels are generated and routing tables constructed for a given set of LSPs in both the schemes.

Let P be the set of loop-free paths for which LSPs are required. Because LSP aggregation is based on destinations, let $P_j \subseteq P$ be the set of paths that have j as the destination. Let a path be represented as a string of node identifiers in which no identifier appears more than once, that is the path is loop free. At router $i \neq j$, for each *unique* subpath $i\beta$ such that $\alpha i\beta \in P_j$ for some subpath α (Note that α can be ϵ), generate a label u that corresponds to subpath $i\beta$. Since $\beta \neq \epsilon$, let $\beta = k\beta'$, and let v be the label for subpath $k\beta'$ at k. Now, a routing table entry of the form $\langle u, k, v\rangle$ is created at router i for u. When router i receives a packet with label u, it replaces it with the label v and forwards it to neighbor k. Fig.1(a)-(b) shows an example. Two paths $abde$ and $acde$ between a and e pass through d. When there is no LSP aggregation, there are two labels u and v at d for these paths. Now using multipath-to-point LSP aggregation, the labels for the two individual LSPs is replaced with a single aggregated label u at d that represents the suffix-path de.

Now consider the two paths $abce$ and $abde$ in Fig.1(c). There are two labels u and v at a, which cannot be aggregated using multipoint-to-point aggregation. This is the limitation of multipoint-to-point aggregation. However using LSMP aggregation, as we will describe shortly, the two paths can aggregated. Consider Fig.1(d), in which when a packet is received with label u, its label can be swapped with any one of the outgoing labels x or z and forwarded to the corresponding neighbors c and d. In LSMP, instead of one next-hop for each incoming label there can be more than one next-hop available with the corresponding outgoing labels. Closely associated with LSMP are *routing parameters* which specify how the next-hops are chosen for packet forwarding. We will describe routing parameters in more detail in the Section 4. We now describe the algorithm for constructing the LSMPs from a given set of paths P. Because the aggregation is based on destination consider the set $P_j \subseteq P$ for a particular destination j. For each destination j, do the following steps.

1. At router $i \neq j$, for each unique prefix subpath α of a path, construct the set of paths $M^i_\alpha = \{i\beta | \alpha i\beta \in P_j\}$. M^i_α is a multipath at i that corresponds to a prefix subpath α. Note that M^i_α can be an empty set. Let $\bar{M}^i_j = \{M_\alpha | \alpha \textit{ is a subpath of some path}\}$. That is, $\bar{M}^i_j$ is the set of unique multipaths at i for j.
2. For each $x \in \bar{M}^i_j$, create a label u_x at i. For each neighbor k, let $y_k = \{k\beta | ik\beta \in x\}$. Now since $y_k \in \bar{M}^k_j$ when $y_k \neq \phi$, let v^k_y be the label for y_k at neighbor k. Then routing table entry $\langle u_x, \{(k, v^k_y) | y_k \neq \phi\}\rangle$ is created at i.

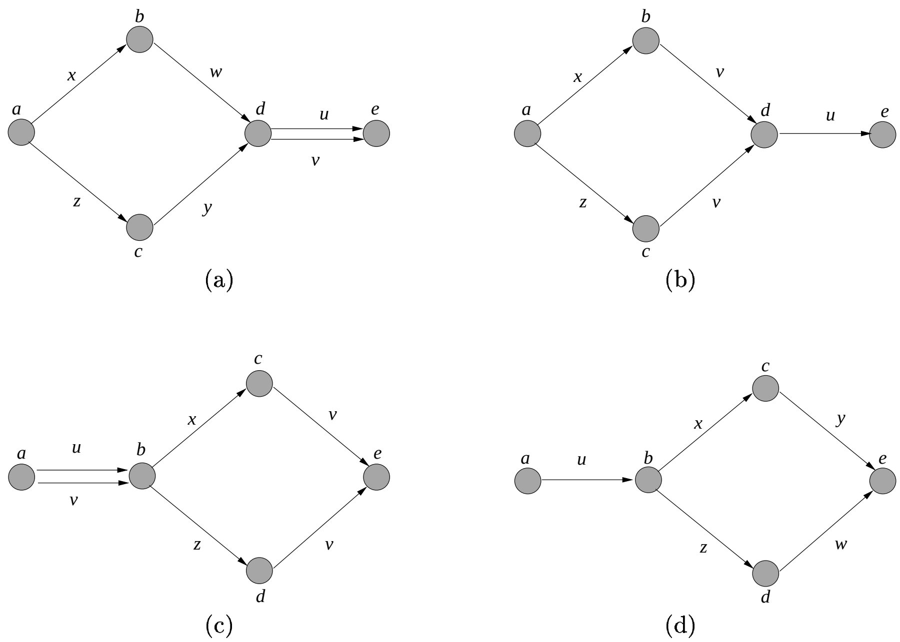

Figure 1. Limitation of Multipath-to-point Aggregation

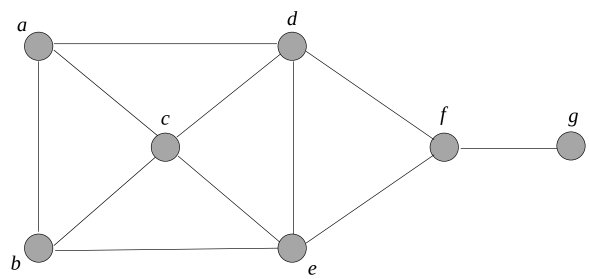

Figure 2. Example illustrating LSMP construction

To illustrate the construction of LSMPs using the above algorithm, consider the network in Fig. 2. Assume the LSMPs for the given set of paths $\langle acdfg\rangle$, $\langle acefg\rangle$, $\langle bcdfg\rangle$ and $\langle bcefg\rangle$ are required to be setup.

We will show how the labels at c is computed. There are two distinct subpaths $\langle a\rangle$ and $\langle b\rangle$ with respect to c and the given set of paths. The multipath associated with subpath $\langle a\rangle$ are $\{\langle cdfg\rangle, \langle cefg\rangle\}$. Similarly the multipath associated with subpath $\langle b\rangle$ is $\{\langle cdfg\rangle, \langle cefg\rangle\}$. Because the two multipaths are identical, only *one* label is created. This label is used by nodes a and b when forwarding packets of destination j to node c. To contrast with other schemes, note that without any LSP aggregation there will be 4 labels and with multipoint-to-point aggregation there will be 2 labels. Now let $\langle acdefg\rangle$ be another path that needs to be setup in addition to the above paths. The LSMP aggregation is as follows. At node c the distinct prefixe paths are again $\langle a\rangle$ and $\langle b\rangle$ and the corresponding multipaths are $\{\langle cdfg\rangle, \langle cefg\rangle, \langle cdefg\rangle\}$ and $\{\langle cdfg\rangle, \langle cefg\rangle\}$. This time the two multipaths are *not equal.* So distinct labels are created at node c for the two multipaths. The label for the first multipath is given to node a and the label for the second multipath is given to b. Note that the label for the first multipath cannot be used by node b because that will allow packets to flow along path $\langle bcdefg\rangle$ which is not a path in the given set. Also note that for the same set of paths, with multipoint-to-point aggregation there will 3 labels and with no LSP aggregation there will be 5 labels.

To make effective use of multiple next-hop choices more than mere specification of next-hop choices is needed. A parameter called *routing parameter* specifying the amount of traffic that must be forwarded to the neighbor must be associated with each next-hop choice. The use of routing parameters is described in the next section. The labels can determined centrally or in a distributed manner. And they can be deposited in the routers using signaling protocols such as LDP or RSVP.

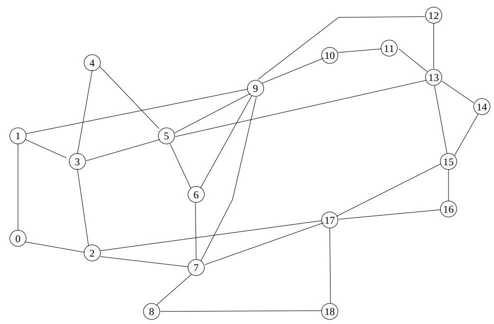

Figure 3. Topology used in the experiment 1

3. EXPERIMENTAL RESULTS

We perform some experiments to measure the effectiveness of LSMP aggregation. The performance metric is the maximum number of labels required in a router for setting up the given set of paths. The maximum number of labels is used because it has direct bearing on the routing table size and complexity of the routing mechanisms. Based on this performance metric, we compare the performance of three schemes: (1) simple LSPs (2) LSP with multipoint-to-point aggregation (MP2P) and (3) LSMP aggregation.

Table 1. Experiment 1 results.

K	NPATHS	LSMP	MP2P	LSP
1	342	74	74	274
2	684	123	166	591
5	1710	165	447	1560
7	2394	181	617	2193
10	3420	188	888	3114
12	4104	195	1045	3740
15	5130	208	1279	4646

In the first experiment, a set of K paths between each source-destination pair of the network shown in Fig.3(a) are randomly generated and setup using the three schemes. The results are shown in Table 1. NPATHS indicates the total number of paths that are setup in each experiment. As can be clearly seen LSMP aggregation offers significant improvement over the other schemes. Also its relative performance gets better as K, and consequently the number of paths NPATHS, increases.

Table 2. Experiment 2 results.

N	NPATHS	LSMP	MP2P	LSP
5	100	19	47	80
7	210	25	80	125
9	360	58	215	320
11	550	85	330	500
13	780	116	465	720
15	1050	151	620	980

In the second experiment, K is kept fixed (K=5), and the size of the network is increased. We use complete graphs of sizes $N = 5, .., 15$. Table 2 shows the results. As can be seen the LSMP aggregation improves over the other schemes as the size of the network grows even when K is kept constant. This is because the total number of paths established in the network continues to grow because the number of source-destination pairs continues to grow. However the rate at when state in the routers grows is slower compared to the rate at which it grows in experiment 1. This is because there are more routers to distribute the paths across.

4. A GUARANTEED SERVICE ARCHITECTURE BASED ON LSMPS

This section describes a QoS architecture based on LSMPs and demonstrates some of its scaling benefits. The main idea in the proposed architecture is that flows are setup and aggregated only along the LSMPs. In the routers, flows are handled only on aggregated basis. By giving delay and throughput guarantees to aggregated flows, the guarantees of individual micro-flows are ensured. In the SMART architecture,[18] we presented special techniques for aggregating flows along multipaths constructed based on distances which we adapt for aggregation flows along LSMPs. We assume that a micro-flow belongs to one of Q flow classes, and that with each class there are a set of paths (say K) between each source-destination pairs that can be used for forwarding packets of that class. The paths are usually determined based on criteria such as QoS, policies and traffic engineering. Using the aggregation technique described in the previous section we construct the LSMPs from the given set of paths; there is one LSMP per-class per-destination. Once the labels that define the LSMPs are obtained, they can be distributed using protocols such as LDP[1] or RSVP.[19]

4.1. Routing Table Structure

To provide delay and throughput guarantees, bandwidth reservations must be made for the flows and schedulers must ensure that the flows receive their share of bandwidth. Because flows are aggregated into classes, routers store link reservations on per-class basis only and links schedule flows on per-class basis. A flow's class is determined at flow setup, and when the application sends its data packets, the ingress router inserts the flow's label in the header of the packet. At the ingress and in the core routers all packets of the same label are merged and handled as a single aggregate flow. The routing table entry for an aggregate flow is as follows:

Incoming label	Destination and class	Incoming Traffic rate	Outgoing labels and Routing Parameters
u	$(j,\ g)$	B	$\{(k, v_k, \phi_k)\}$

where u is the label of incoming traffic of a particular class g and particular destination j. B is the rate of this traffic. For neighbor k, v_k specifies the outgoing label to use if the packet is forwarded to neighbor k, and ϕ_k specifies the fraction of the traffic B that must be forwarded to k. That is, $\phi_k B$ is the amount of traffic of label u that is forwarded to k. The traffic is allocated to the neighbors by a *distributor* according to routing parameters using a weighted deficit round-robin scheme.[18]

4.2. Flow Classes

Using LSMPs requires special methods for merging and splitting aggregate flows that allow deterministic guarantees inspite of aggregation. In the SMART architecture,[18] we presented flow classes that enable such aggregation. For convenience we reproduce some of the results here.

A flow is characterized by the parameters (L, ρ), where L is the maximum size of any packet of the flow and ρ is the average rate of the flow. Flows are grouped into classes based on their maximum packet size and their rate. Assume there are Q real-time classes and with each class g associate a maximum packet size L_g and rate of ρ_g. A flow belongs to class g if the maximum size of its packets is less than L_g and its rate is at least ρ_g.

At the ingress router a flow f with parameters (L, ρ) is shaped with a token bucket with bucket size L and token rate ρ. Now, if the flow belongs to the class g, the delay a packet of the flow experiences at a link (i, k) with propagation delay τ_{ik} and capacity C_{ik} is bounded by $\theta^g_{i,k} = L_g/\rho_g + L_{max}/C_{ik} + \tau_{ik}$, where L_{max} is the maximum packet size allowed for any packet in the network.[18] That is, $\theta^g_{i,k}$ is the delay bound for the class g at link (i, k). We showed that delay bound $\theta^g_{i,k}$ holds for the flow even after it is aggregated with other flows belonging to the same class. The key point is that the delay offered at a link to a flow depends on the class parameters of the flow and link characteristics.

When traffic is allocated to the next-hop neighbors according to routing parameters, an extra burst is introduced due to the non-fluid nature of traffic. The burst is bounded by L_g and can be eliminated using a shaper, but an additional delay of L_g/ρ_g is added to the end-to-end delay. At the link scheduler, there is one queue for each label that has traffic flowing on that link, and the queues are scheduled using a Weighted Fair Queuing (WFQ) scheduler. The per-label scheduling is far more scalable than per-flow scheduling because of the amount of label reduction that is achieved using the LSMP aggregation. The link scheduler introduces some delay-jitter in the traffic passing through the link. This is removed at the receiving router through shaping, again using a token bucket, before it is merged with other flows. The traffic emerging from the shaper belongs to class g and can be readily merged with the shaped traffic with the same label received on the other links. The shaping mechanism for aggregated flows is described in[18] Thus the end-to-end delay of a flow is the total delay on the longest delay path from the source to the destination on the LSMPs. The delay of a path is the sum of the delays of each link on the path and the delay experience at each distributor on the path. This delay can be computed recursively based on network parameters and class parameters as follows. Let δ^i_u represents the end-to-end delay for traffic that router i receives with label

u, and let g be the class of this flow. Also let S_u^i be the set of next-hop choices in the routing table entry for u at i. Then the end-to-end delay is defined by

$$\delta_u^i \quad = \quad \frac{L_g}{\rho_g} + MAX\{\theta_{i,k}^g + \delta_{v_k}^k \mid k \in S_u^i\}$$

where v_k is the outgoing label for u on link to neighbor k. Note that $\delta_{j,g}^j = 0$ for all j and g. Because the class and destination of a flow are known at the ingress rouer, the end-to-end delay for a flow is available a priori even before the flow is setup.

4.3. Signaling

When a request arrives, a path is selected at the source based on heuristics such as *widest shortest path*(WSP).[13] To use the heuristic, however, the link states must be advertised regularly resulting in substantial overhead.[7] Using LSMPs, however, the use of bandwidth advertisements can be eliminated without much loss of performance. It is based on the observation that delay and bandwidth constraints depend on different time scales and as such they must be decoupled from each other. The LSMPS allow this decoupling because LSMPs can established a priori based on static constraints such as delay (and jitter), while the bandwidth along the LSMPs can allocated at the time of flow request.

As there is more than one next hop available, at each hop *the widest outgoing link* can be chosen. We showed in[16] that this simple heuristic offers performance comparable to heuristics such as WSP that use bandwidth advertisements. The LSMP approach is different from the k-shortest path approach in that the complete path is committed at the source of the request in the k-shortest path approach. This requires available bandwidth information of non-adjacent links. And committing the complete path based on local information can result in high call-blocking rates. Heuristics based on localized information[11] can be used instead, but they can be computation intensive as they involve on-line measurements.

When flows are signaled through the network, the routing parameters are adjusted to reflect the reservations. When a flow request for a destination of certain bandwidth r and class g arrives at the source node, the source node first obtains the label u and the corresponding routing table entry for this destination and class. The source node then initiates a reservation request of form $REQ(u, r)$. When a router receives a request of the form $REQ(u, r)$, it finds the *widest link* among the outgoing links for that label u. If k is the neighbor that corresponds to this link, let v_k and ϕ_k be the outgoing label and the routing parameter respectively. The bandwidth B is incremented by r and the routing parameter ϕ_k is modified appropriately.[16] A request $REQ(v_k, r)$ is then forwarded to k. This process is repeated until the request reaches the destination. If the request failed at any hop, the flow setup is considered to have failed and reservations are released.

When flows terminate, the routing tables are similarly modified. The source of the flow issues a message $REL(u, r)$ to the ingress router. When a routers receives a release message, the corresponding bandwidth rate B is decremented by r and the routing parameters ϕ_k is modified accordingly as described in.[16] The consistency of aggregated reservations can be maintained using soft-state mechanism similar to the AGREE protocol.[18] The refresh messages are on per-label basis. Because the number of labels in the LSMP are significantly less than those in per-flow approaches, the resulting soft-state refresh mechanism is relatively much more scalable.

The hop-by-hop admission control described above uses only local information (available bandwidth on the outgoing links) while in the k-shortest path approach, the complete path is committed at the source based on available link bandwidth information which is often outdated because of the latencies inherent in any advertising scheme. This leads to higher call-blocking rates. In contrast, in our scheme, the decision of including a link in the path is delayed until the request arrives at the head of the link. The heuristic used in choosing the outgoing link is based on information that is very accurate thus contributing to the performance of the scheme. This is a topic that is beyond the scope of this paper and is discussed in detail in another publication.[16]

A drawback of the architecture is that it does not allow delay as one of the constraints specified by the client in this model. The client only specifies the required throughput and the characteristics of the flow. The delay is specified by the network based on the parameters specified by the client. Another drawback is that packets can arrive out-of-order at the destination due to the use of multipaths, but this should not pose a problem when end-to-end delay bounds are simultaneously provided. Using the delay bounds the out-of-order packets can be easily reordered.

5. CONCLUSIONS

We described a scalable QoS architecture based on the MPLS technology that uses highly aggregated state in the routers, yet provides deterministic delay and bandwidth guarantees. We introduced the notion of *label-switched multipath* (LSMP) and described how flows can be aggregated along the LSMPs. To address the problem of providing deterministic guarantees in the presence of flow aggregation, a special technique for defining and using flow classes has been employed. The suggested aggregation schemes significantly reduce the amount of state that needs to be maintained in the routers making the architecture and the associated signaling protocols scalable. The LSMP aggregation is more powerful than the well-known multipoint-to-point aggregation and can also be used in other contexts such as Traffic Engineering and Differential Services architectures.

REFERENCES

1. O. Aboul-Magd and et al. Constraint-Based LSP Setup using LDP. *Internet Draft, draft-ietf-mpls-cr-ldp-05.txt*, February 2001.
2. D. Awduche and et al. A Framework for Internet Traffic Engineering. *Internet Draft, draft-ietf-tewg-framework-00.txt*, January 2000.
3. S. Berson and S. Vincent. Aggregation of internet integrated services state. *IWQOS*, 1998.
4. R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: An overview. *RFC 1633*, June 1994.
5. R. Callon and A. Viswanathan et. el. A framework for multiprotocol label switching. *Internet Draft*, Nov. 1997.
6. D. Black et al. An Achitecture for Differentiated Services. *Internet Draft*, May 1998.
7. G. Apostolopoulos et al. Quality of Serive Based Routing: A Performance Perspective. *Proc. of ACM SIGCOMM*, 1998.
8. R. Guerin et al. Scalable QoS Provision Through Buffer Management. *Proc. of ACM SIGCOMM*, 1998.
9. Rampal et al. Flow grouping for reducing reservation requirements for guaranteed delay service. *Internet Draft:draft-rampal-flow-delay-service-01.txt*, July 1997.
10. Y. Bernet et al. An Framework for Differentiated Services. *Internet Draft*, May 1998.
11. S. Nelakuditi et.al. Adaptive proportinal routing: A localized qos routing approach. *Proc. IEEE INFOCOM*, 2000.
12. D. Ferrari, A. Benerjea, and H. Zhang. Network support for multimedia - a discussion of tenet approach. *Computer Networks and ISDN*, 26:1267–1280, 1994.
13. Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. *Proc. International Conference on Network Protocols*, October 1997.
14. E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. *Internet Draft*, March 1997.
15. A. Viswanathan, N. Feldman, and et. al. Aris: Aggregate route-based ip switching. *Internet Draft*, March 1997.
16. S. Vutukury and J.J. Garcia-Luna-Aceves. A Scalable Architecture for Providing Deterministic Guarantees. *Proc. of ICCCN*, Oct. 1999.
17. S. Vutukury and J.J. Garcia-Luna-Aceves. A Multipath Framework Architecture for Integrated Services. *Proc. IEEE GLOBECOM*, 2000.
18. S. Vutukury and J.J. Garcia-Luna-Aceves. SMART: A Scalable Architecture for Intra-domain QoS Provisioning. *QOS-IP 2001, International Workshop on QoS in Multiservice IP networks*, 2001.
19. L. Zhang and et al. RSVP: A New Resource Reservation Protocol. *IEEE Communications Magazine*, 31(9):8–18, 1993.